The 1st International Student Conference on Economics and Business Excellence (ISCEBE) 2024
e-ISSN:xxxx-xxxx/Vol. 1 No.1 (2024)

THE IMPACT OF SYSTEM DEVELOPMENT LIFE CYCLE IMPLEMENTATION ON SOFTWARE PRODUCT QUALITY: A SYSTEMATIC LITERATURE REVIEW

Ulfa Annisa[1)], Asniati Bahari[2)]
[1,2)]Accounting Department, Andalas University, Indonesia
Corresponding author: asniati@eb.unand.ac.id

Abstract

The demand for high-quality software has grown over the years, especially in recent times with the digital explosion. Only efficient development methodology can give the best result possible. This review focuses on the impacts of SDLC implementation on software quality from the perspective of a literature review and meta- analysis, using related articles from 2014 to 2025. A total of 11,851 identified papers were assessed, out of which the retrieved documents went through a series of eliminations until only 10 relevant studies remained. It generally examines the benefits, disadvantages, and challenges generally faced in the industry through a comparison of modern models such as Agile and DevOps with traditional models such as Waterfall and Iterative. It was found that existing SDLC models like Agile and DevOps are flexible, highly contributive by teams, and work in a development cycle effectively. Clearly, they have a specified rate of delivery, easy adjustments to changes in needs, and feedback mechanisms. Problems noted are: not having a form of documentation, managing a large project is cumbersome, and technology changes so rapidly. Although structured and applicable to projects with fixed requirements like Waterfall and Iterative, it is very difficult to achieve the level of speed and flexibility reached by modern models. According to the research work, modern models suit dynamic environments, though the ultimate choice depends on project needs, the nature of the team, and organization culture. The aim of this section is to provide strategic insights into the different SDLC models, relative advantages, and disadvantages that will enable software developers to make an informed choice of the best model that will yield high-quality software capable of adapting to the dynamic demands of users and changing industry expectations.

Keywords: *System Development Life Cycle (SDLC), Agile, DevOps, software quality, systematic literature review*

Introduction

The effective development techniques are required for achieving the best outcomes because high-quality software is in great demand in the digital era. Therefore, the current study was performed based on a critical literature review and meta-analysis of ten papers that were published from 2014 to 2025 to evaluate the effect of SDLC implementation on software quality. With the initial identification of 11,851 papers, these selected articles underwent a strenuous filtering process entailing multiple steps of refinement, reducing it down to only 10 relevant studies. This work draws on many advantages, disadvantages, and challenges evident within the industry, comparing some of the modern models like Agile and DevOps against their more classic versions like the Waterfall and Iterative models. However, because Agile and DevOps frameworks are more flexible, development teams can respond swiftly to shifting consumer and market expectations (Pargaonkar, 2023). It has also been shown that modern SDLC methods, especially Agile, have faster development cycles, are more team-oriented, and present fewer project risks. This flexibility should facilitate finding and rectifying problems well in time, which is a radical improvement in software quality. Research has shown that the faster a problem is resolved during the development process, the less time and money will be required for changes later on, thus leading to an effective and generally higher-quality final product (Ergasheva & Kruglov, 2020). Adopting SDLC is not without its challenges. A major problem arises in adjusting to the changing user requirements, especially in the case of traditional models such as Waterfall. The rigid nature of this model often leads to increased cost and longer duration of time if the initial planning is not good enough. Approaches like Agile and DevOps solve this challenge by introducing flexibility in every cycle of development, hence proving better in maintaining high software quality (Agarwal et al., 2023).

Understanding how SDLC implementation affects the quality of software products becomes critical as advances in technology continue to create higher expectations within the industry. Looking into different models of the SDLC provides valuable details on how each one will impact software quality. DevOps, for instance, centers on automation and continuous integration in an effort to accelerate development speed and maintain quality standards (Akbar et al., 2018). However, most of the current literature concentrates on specific SDLC models and not on general impacts of the implementation of SDLC in software quality. This thesis proposal will, therefore, be premised on an intensive literature review to establish a critical deterrent to SDLC implementation in an attempt to address the above-identified literature gap. In support of the above thesis objective, the following questions shall be addressed:

RQ1: How does the implementation of SDLC impact software quality?
RQ2: What are the main challenges in implementing SDLC, and how do they affect product quality?
RQ3: How does the SDLC methodology influence the effectiveness of software quality?

This research seeks to provide a comprehensive understanding of the relationship between SDLC implementation and software product quality, contributing valuable insights to both academic and practical applications.

Pargaonkar (2023) positions the modern methodologies of Agile and DevOps against traditional SDLC models such as the Waterfall, Iterative, and V-Model. The waterfall model is characterized by structured, linear, and well-documented procedures that provide better project monitoring but less flexibility in case of unforeseen changes. Agile and DevOps emphasize flexibility, quick iterations, collaboration, and a more rapid adjustment to changes. However, in later stages of development, the reduced focus on documentation by Agile might compromise software quality (Pargaonkar, 2023). In projects where needs are clearly defined and regulatory limits are strict, compliance and mitigation of risks are paramount; hence, waterfall methodology still applies. Its inflexibility has however given way to flexible models that allow input in every stage of the project life cycle, such as Iterative and Spiral (Gupta, 2021).

Keeping in view the developments in the technologies and steadily rising demands by the users, many new SDLC models have been developed that complement the shortcomings of conventional methods. One such model is the AZ-Model proposed by Akbar et al. in 2018. In this paradigm, some additional tasks are added in each stage of SDLC to upgrade the quality of the output software. It differs from traditional methods, such as the Waterfall model, in offering superior error management, shorter cycles of projects, and quality regarding software.

While modern SDLC models, such as Agile and DevOps, have their merits in terms of flexibility and adaptability, they also introduce some challenges. The key challenge is how to balance rapid iteration cycles against thorough documentation, which starts to become quite critical in regard to effective communication and tracking. Moving from a traditional model, such as Waterfall, into a more dynamic model disrupts teams that are set in clear-cut processes. Not having any clear guidelines often results in inconsistent results with some modern models if not managed properly.

Thus, the quality of the resultant software will depend on a series of events based on what methodology the SDLC focuses on. Agile, while incorporating feedback and collaboration right from the beginning, increases the likelihood of higher software quality because defects can be resolved early, whereas traditional ones, like Waterfall, might encounter problems in regard to the quality if alterations take place a little later into the project. The capacity to incorporate user feedback and adjust to changing requirements is crucial in enhancing the final product, especially in dynamic environments (Ergasheva & Kruglov, 2020).

Methods

The literature search was carried out in three steps: First, the PICOC framework was used: Population, Intervention, Context, Outcome, and Comparison-to break down the essential terms of each research question. Next, we consulted field experts and performed a dictionary search for synonyms of the keywords. We combined the synonyms using "OR" and linked the elements of the PICOC structure with "AND," including the wildcard "*.". After the identification of the keywords, refinement through the syntax requirements of the search engine was performed. All adjustments were carefully recorded in a protocol document.

Research Questions (RQ):
RQ1: How does the implementation of SDLC impact software quality?
RQ2: What are the main challenges in implementing SDLC, and how do they affect product quality?
RQ3: How does the SDLC methodology influence the effectiveness of software quality?

Example of the search process for RQ1:
Step 1: PICOC Structure
RQ1: What is the impact of implementing SDLC on software quality?
Population: Software development, SDLC
Intervention: Implementation, methods, models

Outcome: Quality, performance, defects, user satisfaction
Comparison: Traditional SDLC versus modern SDLC models

Step 2: Synonyms
Population: “Software development,” “SDLC,” “System development”
Intervention: “Implementation,” “Adoption,” “Method,” “Framework,” “Model”
Outcome: “Quality,” “Product performance,” “Defects,” “User satisfaction”

Step 3: OR, AND, Wildcards
Search string: ("Software development") AND ("SDLC" OR "System development" OR "Software methodology") AND ("Implementation" OR "Adoption" OR "Model") AND ("Quality" OR "Product performance" OR "Defects")

Materials would, therefore, be retrieved only from three major databases-ScienceDirect, Emerald, and Google Scholar-selected in view of their rich library collection relating to the subjects under review in software development, SDLC, and quality management. Furthermore, it shall consider only materials published within a timeline of 2014 through 2025. All materials reviewed were to be a peer-reviewed journal article, conference paper, or book chapters.

The inclusion and exclusion criteria used to ensure that only relevant studies were considered included the following:

Inclusion Criteria:

•The focus of the study will be on the implementation of SDLC in software development.
•The research must talk about software quality, product performance, or user satisfaction.
•The article is to be published within 2014 and 2025.
•Articles must be available as full text.

Exclusion Criteria:

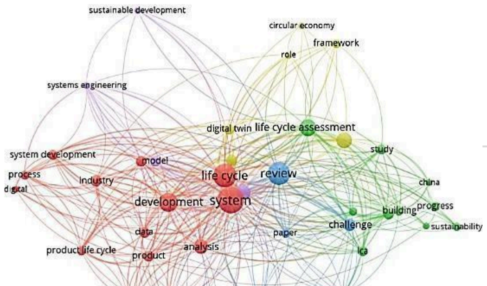
•Research that is not focused on SDLC or software quality.
•Non-peer-reviewed articles.
•Final Selection Process.

In this study, after applying the inclusion and exclusion criteria to the results from the search, a total of 10 articles were compiled. These articles were further critically analyzed for relevance and quality in relation to the contribution each made toward the research questions. This filtering led to the final selection of articles summarized in Table 1 below, showing the number of articles chosen from each database.

Tabel 1 Database Repositories & Results of Selection Processes

Database	URL	No Filter	1st Filter	2nd Filter	3rd Filter
Science Direct	https://www.sciencedirect.com/	2431	121	37	3
Emerald Insight	https://www.emerald.com/insight/	1877	87	25	2
Google Scholar	https://scholar.google.com/	7543	303	51	5
	Total	11851	511	113	10

A mapping exercise was done that focused on the implementation of the SDLC and its impacts on the quality of software products to lay a strong foundation for this research. This mapping was done to find trends and clusters of keywords related to the subject of research. The mapping was supported using the tools Publish or Perish and VOSviewer. Searches were made in journals by using the ScienceDirect, Emerald, and Google Scholar databases through Publish or Perish. The keywords involved in the search included "System Development Life Cycle," "SDLC," "Software Quality," and "Software Development Models." A total of 1,043 articles were gathered from Google Scholar, 132 from ScienceDirect, and 98 from Emerald. The data were then analyzed and visualized by using VOSviewer. The resulting keyword clusters are shown in Figure 1.



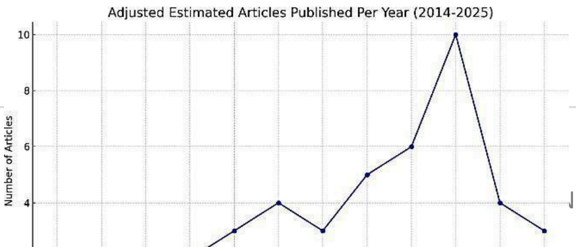


Fig 1. (a) Research Mapping; (b) Article in Years.

Figure 1.(a) presents a map of keywords of articles corresponding to SDLC and software product quality. The map represents six clusters visually separated by color. The VOSviewer defines a cluster as a set of terms frequently mentioned together in the literature. The terms have been assigned to clusters based on their co-occurrence relationships and the colors refer to assignment to the respective cluster. Figure 1(a) shows that the terms "life cycle" and "development" are outstanding because they have been used 95 and 87 times, correspondingly, to underline the central role in this research area, whereas "quality" was mentioned only 30 times, underlining that SDLC is so well explored although its connection with software quality remains not well investigated. Figure 1. b shows the number of articles published between 2014 and 2025. From this trend, it is observed that research is gradually growing from year to year and had a spike in the year 2023 with peak publications. The peak marks growing interest in the study of SDLC and its applications probably due to modern software development methodologies like Agile and DevOps. Other fluctuations over the years reflect shifting research emphases and changes in the field, but they serve to indicate that the subject remains one of considerable scholarly importance.

Result and Discussions

Table 2 summarizes the key aspects of SDLC implementation and its impact on software product quality, highlighting the main findings from the literature review.

Tabel 2. Aspects of SDLC Implementation and Their Influence on Software Quality: A Systematic Review

Research Question	Aspects of Discussion	Source
RQ1: How does the implementation of SDLC impact software quality?	Impact on Product Performance and Functionality, Relationship of SDLC Stages with Software Quality Attributes (e.g., Reliability, Maintainability, Usability) and The role of documentation and requirements gathering in the context of ensuring software quality.	Siregar (2020, Notarnicola et al. (2017), Wahid (2020)
RQ2: What are the main challenges in implementing SDLC, and how do they affect product quality?	Identification of Common Challenges During the Implementation of SDLC: Time Constraints, Resource Limitations, Scope Creep, Among Others, Effects of Poor Implementation on Software Defects and User Satisfaction and Risk Management in SDLC: Its Influence on Product Quality	Maulana (2022), Pargaonkar (2023), Maukar & Irwansyah (2024)
RQ3: How does the SDLC methodology influence the effectiveness of software quality?	Comparison of various SDLC methodologies regarding quality outcomes, such as Waterfall, Agile, and DevOps, How iterative or incremental approaches influence continuous improvement in quality and The implication of testing phases in various SDLC models on defect reduction.	Purba (2021), Maukar & Irwansyah (2024), Singh (2018)

The table gives a detailed summary of the important factors relating to the SDLC process and its impacts on software product quality. It indicates the major issues present during the implementation of SDLC, how different SDLC methodologies- Agile, Waterfall, DevOps-affect the quality of the software, and how these different methodologies will either reduce or increase defects, performance, and maintainability. These insights are taken from a critical review of the literature, which has emphasized the complicated relationship between SDLC processes and the resultant quality of software.

1. RQ1: How Does the Implementation of SDLC Impact Software Product Quality?

It will now discuss the importance of using the System Development Life Cycle. SDLC models, Wahid 2020 elaborates, adopt the Waterfall model and its variant, providing a planned systematic and step-by-step procedure in which every step in the planning, design, implementation, testing, deployment, and maintenance needs to be completed before going further in sequence. This is a process wherein each stage is highly scrutinized for its results and produces more reliable, high-quality software. Siregar (2020) insists on the necessity of security testing right through each stage of SDLC. In this way, as a security concern is detected upstream of the process, the chances that there will be defects or loopholes are reduced, so one can get more stable and secure software. Apart from that, Notarnicola et al. (2017) discuss various structured lifecycle methodologies about LCA on software. Just like the SDLC, this model brings better stability of the product, ensures environmental sustainability, and gives a long time for efficiency. Finally, effective implementation of SDLC delivers better quality of the software with phase-by-phase processing and detailed testing that incorporates sustainability at each step of the development.

2. RQ2: What Are the Main Challenges in Implementing SDLC, and How Do They Affect Product Quality?

SDLC implementation is characterized by different challenges that can directly affect the quality of software products. According to Maulana (2022), one of the major challenges is reliance on traditional methods of promoting products and managing non-digital data. This mostly makes the process slower and error-prone, especially in areas such as inventory management and sales reporting. Maukar and Irwansyah (2024) further enumerate the disadvantages of using Microsoft Excel manually while performing data handling: human error, lower accuracy of the data, and some specific defects in security. After all, all these factors could provide an impact on the result with respect to the product's quality. Pargaonkar (2023) provides further weight to limits created by traditional SDLC models themselves-a Waterfall model that has no flexibility. This might be a big minus for those projects whose requirements keep on continuously evolving, since once a phase is complete, going back to it again-in case some changes need to be made-is pretty difficult. On the contrary, Agile methodologies have more flexibility but bring in challenges related to documentation and managing large-scale projects. These factors affect the quality of the product if not carefully managed. It hence challenges and optimizes the quality of the software product to choose the most appropriate model for every project.

3. RQ3: How Does the SDLC Methodology Influence the Effectiveness of Software Quality?

SDLC methodology has contributed significantly to the improvement of software quality and efficiency, since it provides a systematic approach toward development. In the research study by Silitonga and Purba (2021), the SDLC model was followed for the design of a web-based hospital patient registration system. Its implementation reduced queues while improving user satisfaction since it automated the administrative task and provided a user-friendly interface. Results indicated that SDLC reduced errors in data processing, therefore increasing the quality of service delivery. In this respect, the research of Maukar and Irwansyah 2024 has identified the advantage of using SDLC in designing business intelligence dashboards of the toy manufacturing company. With this SDLC, smooth identification of issues related to the previous system was found to create a more productive design of data architecture by enabling tools like Microsoft Power BI. Thus, with results, decision-making could be swift as well as accurate and resource usage optimally utilized.

Maulana, in his 2022 study, explained the application of the Waterfall model in an e-commerce system for smartphone sales. He found that structured phases of SDLC improve effectiveness in digital promotions and stock management, replacing error-prone manual processes. Thereafter, the company may achieve better sales outcomes or operational efficiency.

A 2015 study on SDLC application in the web-based sales system of batik products reported that SDLC has addressed limited marketing reach and data security. The system developed with increased market reach through online platforms, easy customer transaction, and positively impacted the sales and corporate reputation of batik. From the analyzed studies, it was concluded that SDLC methodology enhances the quality of the software, meeting user needs and ensuring its long-term reliability.

4. RQ3: How Does the SDLC Methodology Influence the Effectiveness of Software Quality?

SDLC methodology has contributed significantly to the improvement of software quality and efficiency, since it provides a systematic approach toward development. In the research study by Silitonga and Purba (2021), the SDLC model was followed for the design of a web-based hospital patient registration system. Its implementation reduced queues while improving user satisfaction since it automated the administrative task and provided a user-friendly interface. Results indicated that SDLC reduced errors in data processing, therefore increasing the quality of service delivery. In this respect, the research of Maukar and Irwansyah 2024 has identified the advantage of using SDLC in designing business intelligence dashboards of the toy manufacturing company. With this SDLC, smooth identification of issues related to the previous system was found to create a more productive design of data architecture by enabling tools like Microsoft Power BI. Thus, with results, decision-making could be swift as well as accurate and resource usage optimally utilized.

Maulana, in his 2022 study, explained the application of the Waterfall model in an e-commerce system for smartphone sales. He found that structured phases of SDLC improve effectiveness in digital promotions and stock management, replacing error-prone manual processes. Thereafter, the company may achieve better sales outcomes or operational efficiency.

A 2015 study on SDLC application in the web-based sales system of batik products reported that SDLC has addressed limited marketing reach and data security. The system developed with increased market reach through online platforms, easy customer transaction, and positively impacted the sales and corporate reputation of batik. From the analyzed studies, it was concluded that SDLC methodology enhances the

quality of the software, meeting user needs and ensuring its long-term reliability.

Conclusions

This review explains how SDLC methodologies have influenced software product quality through the year-by-year, step-by-step analysis of selected ten papers from 2014 to 2025. From the results, it was found that software quality had increased notably due to the fact that the SDLC provided a systematic way to develop software. Agility and DevOps contributed to bringing flexibility, collaboration within the teams, and adaptation to changing requirements directly into the modern model for contributions toward the quality of a product. Traditional models, such as the Waterfall, are pretty well-structured and sometimes quite reliable in projects operating on fixed requirements, but such models lack the flexibility that is needed for any dynamic environment. Even with the advantages, some of the challenges that SDLC methodologies face include a lack of documentation in Agile, rigidity in traditional models, and problems with scaling on large projects. Overcoming these challenges means assuring software quality and catering to the evolving needs of users. Selection of methodology in SDLC is very crucial to enhancing development efficiency and effectiveness. Agile and DevOps-enabled projects shrink iteration cycles, cope with errors in a more constructive way, use fewer resources, and increase throughput. Traditional models find their best application for projects that require high regulation.

Some of the limitations in this review are that the researcher focused on only three major databases: ScienceDirect, Emerald, and Google Scholar. This may not have captured all relevant studies from other databases. The study also focused on English language peer-reviewed articles; hence, many other informative works from non-English literature may not have been captured. This might be further limited by the data availability of emerging methodologies of SDLC. Future studies should research more types of databases and languages to facilitate the gathering of diverse responses to questions regarding SDLC implementation. Other hybrid SDLC models-like combining elements of Agile and Waterfall to meet particular project needs-need to be explored in further research with respect to their impacts. Furthermore, other welcome topics are investigations into the roles of emerging technologies like AI and machine learning in improving SDLC processes, and longitudinal studies on long-term effects created by SDLC methodologies in software maintainability and performance. These will go a long way toward a better understanding of how to optimize SDLC methodologies for enhanced software quality and adaptability within an ever-changing technological environment.

References

Agarwal, A., Agarwal, A., Verma, D., Tiwari, D., & Pandey, R. (2023). A Review on Software Development Life Cycle. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. https://doi.org/10.32628/cseit2390387

Akbar, M., Sang, J., Khan, A., Fazal-E-Amin, .., Nasrullah, .., Shafiq, M., Hussain, S., Hu, H., Elahi, M., & Xiang, H. (2018). Improving the Quality of Software Development Process by Introducing a New Methodology–AZ-Model. IEEE Access, 6, 4811-4823. https://doi.org/10.1109/ACCESS.2017.2787981.

Aminu, H., & Ogwueleka, F. (2020). A Comparative study of System Development Life Cycle Models. Journal of emerging technologies and innovative research.

Ergasheva, S., & Kruglov, A. (2020). Software Development Life Cycle early phases and quality metrics: A Systematic Literature Review. Journal of Physics: Conference Series, 1694. https://doi.org/10.1088/1742-6596/1694/1/012007.

Maukar, A., & Irwansyah, M. A. (2023). Developing a business intelligence dashboard of liquid material at a toy manufacturing company using a System Development Life Cycle (SDLC) model. *Inform: Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, *9*(1), 32–41. https://doi.org/10.25139/inform.v9i1.6705

Maulana, I. T. (2022). Penerapan metode SDLC (System Development Life Cycle) Waterfall pada e-commerce smartphone. *Jurnal Ilmiah Sistem Informasi dan Ilmu Komputer*, *2*(2), Article 162. https://doi.org/10.55606/juisik.v2i2.162

Notarnicola, B., Sala, S., Anton, A., McLaren, S. J., Saouter, E., & Sonesson, U. (2017). The role of life cycle assessment in supporting sustainable agri-food systems: A review of the challenges. *Journal of Cleaner Production*, *140*(Part 2), 399–409. https://doi.org/10.1016/j.jclepro.2016.06.071

Pargaonkar, S. (2023). A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering. International Journal of Scientific and Research Publications.

https://doi.org/10.29322/ijsrp.13.08.2023.p14015.

Silitonga, P. D. P., & Purba, D. E. R. (2021). Implementasi system development life cycle pada rancang bangun sistem pendaftaran pasien berbasis web. *Jurnal Sistem Informasi Kaputama (JSIK)*, 5(2), 196–203

Singh, A., & Kaur, P. (2018). Analysis of Software Development Life Cycle Models. Lecture Notes in Electrical Engineering. https://doi.org/10.1007/978-981-10-8234-4_55.

Wahid, A. A. (2020). Analisis metode Waterfall untuk pengembangan sistem informasi. *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*, *Oktober*(1), 1–12.Aminu, H., & Ogwueleka, F. (2020). A Comparative study of System Development Life Cycle Models. Journal of emerging technologies and innovative research.

Ergasheva, S., & Kruglov, A. (2020). Software Development Life Cycle early phases and quality metrics: A Systematic Literature Review. Journal of Physics: Conference Series, 1694. https://doi.org/10.1088/1742-6596/1694/1/012007.

Maukar, A., & Irwansyah, M. A. (2023). Developing a business intelligence dashboard of liquid material at a toy manufacturing company using a System Development Life Cycle (SDLC) model. *Inform: Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, *9*(1), 32–41. https://doi.org/10.25139/inform.v9i1.6705

Maulana, I. T. (2022). Penerapan metode SDLC (System Development Life Cycle) Waterfall pada e-commerce smartphone. *Jurnal Ilmiah Sistem Informasi dan Ilmu Komputer*, *2*(2), Article 162. https://doi.org/10.55606/juisik.v2i2.162

Notarnicola, B., Sala, S., Anton, A., McLaren, S. J., Saouter, E., & Sonesson, U. (2017). The role of life cycle assessment in supporting sustainable agri-food systems: A review of the challenges. *Journal of Cleaner Production*, *140*(Part 2), 399–409. https://doi.org/10.1016/j.jclepro.2016.06.071

Pargaonkar, S. (2023). A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering. International Journal of Scientific and Research Publications. https://doi.org/10.29322/ijsrp.13.08.2023.p14015.

Silitonga, P. D. P., & Purba, D. E. R. (2021). Implementasi system development life cycle pada rancang bangun sistem pendaftaran pasien berbasis web. *Jurnal Sistem Informasi Kaputama (JSIK)*, 5(2), 196–203

Singh, A., & Kaur, P. (2018). Analysis of Software Development Life Cycle Models. Lecture Notes in Electrical Engineering. https://doi.org/10.1007/978-981-10-8234-4_55.

Wahid, A. A. (2020). Analisis metode Waterfall untuk pengembangan sistem informasi. *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*, *Oktober*(1), 1–12.